CLAIMS

What is claimed is:

1. A computer system comprising:

a storage device coupled to a processor and having stored therein at least one routine, which when executed by the processor, causes the processor to generate data, the routine at least causing the processor to at least,

generate at least one test program; and

analyze the test program; and

generate at least one subsequent test program until at least one termination criterion is met.

2. The computer system of claim 1, further comprising:

a population of data stored in a storage device.

3. The computer system of claim 2, wherein a portion of the population is replaced.

4. A machine readable storage medium containing executable program instructions which when executed cause a digital processing system to perform a method comprising:

(a) generating a test program;

(b) evaluating the test program based upon coverage data;

(c) using the evaluation to select a new test program.

5. The computer system of claim 3, further comprising:

(d) determining whether a population has reached a desired size of the population.

6. The computer system of claim 3, wherein the population has not reached the desired size, the method further comprising:

(e) creating an abstract representation of a functional test program.

7. The computer system of claim 6, wherein the abstract representation is translated into a functional test program.

8. The computer system of claim 7, further comprising:

(f) executing at least one test program; and

(g) generating coverage data.

9. The computer system of claim 8, further comprising:

(h) storing abstract representation and corresponding coverage data into a storage device.

10. The computer system of claim 9, wherein if desired coverage has not been achieved, operations (a) through (h) are repeated.

11. A method comprising the computer-implemented operations of:

determining a population size, a first logic which if the population has not reached a designated size, then a representation of a test program is randomly generated, a second logic which if the population has reached a designated size, then a genetic operation is chosen and select at least one test program from a population; and

modify the test program(s) using the genetic operation to create at least one new test program;

executing the new test program(s);

measuring coverage data; and

placing the new test program(s) and coverage data into the population.

12. The method of claim 11, wherein the genetic operation is a mutation; and

choosing one test program based upon coverage.

13. The method of claim 12, further comprising:

replacing a portion of the population.

14. The method of claim 13, wherein the genetic operation is a crossover operation, and

choosing two test programs.

15. The method of claim 14 further comprising:

performing a crossover operation.

16. A computer implemented method comprising:

determining a population size, a first logic that when population size has not attained its desired size, then an empty abstract syntax tree is created and a second logic that if the population has attained its desired size then a genetic operation is chosen,

filling the abstract syntax tree with application-specific node types;

translating the abstract syntax tree into an application-specific test program;

executing the test program by the computer processor and generating coverage data; and

placing the abstract syntax tree and corresponding coverage data into a population.

17. The method of claim 16, wherein choosing a genetic operation further comprises:

choosing a mutation operation.

18. The method of claim 17, further comprises:

choosing at least one abstract syntax tree, and

replacing a portion of the coverage data.

19. The method of claim 17, further comprising:

choosing at least two abstract syntax trees; and

performing a crossover operation.

20. A computer implemented method comprising:

determining a population size;

choosing a genetic operation;

choosing two abstract syntax trees based upon coverage data;

performing a genetic operation to form a new abstract syntax tree;

translating the abstract syntax tree into an application-specific test program;

executing the test program;

generating coverage data; and

putting the abstract syntax tree and corresponding coverage data into a population.